

Csoport neve: The Racoons

Feladat sorszáma: H\_4

Feladat címe: Rendszertervezés

## **Clean Paws Library elektronikus könyvtár**

Gyakorlatvezető:

Árvai László

Csoport tagok:

Árvai László	TANKÖR	<a href="mailto:laszlo.arvai@uni-miskolc.hu">laszlo.arvai@uni-miskolc.hu</a>
Kovács Dániel	F9Y7TW	<a href="mailto:kovacsd435@gmail.com">kovacsd435@gmail.com</a>
Mészáros István	WZZTLL	<a href="mailto:mesziist@gmail.com">mesziist@gmail.com</a>
Mester Dániel	FZIE3D	<a href="mailto:danielmester22@gmail.com">danielmester22@gmail.com</a>
Fónagy Zeus Vilmos	G6HV2W	<a href="mailto:fonagyzeusz02@gmail.com">fonagyzeusz02@gmail.com</a>
Kiss Tamás László	WN6YVX	<a href="mailto:tamaskiss2003@gmail.com">tamaskiss2003@gmail.com</a>

**BEADÁS DÁTUMA**

## Történet

Dátum	Verzió	Leírás	Szerző
2023.11.19.	0.0.1	A dokumentum elkezdése	Kovács Dániel
2023.11.21	0.1.0	A dokumentum folytatása	Kovács Dániel
2023.11.22	1.0.0	Dokumentum Befejezése	Kovács Dániel
2023.12.12	1.0.1	Relációs modell update	Mészáros István

# 1. Bevezetés

Ebben a dokumentumban részletesen kifejtjük a Clean Paws Library (CPL) mögött álló adatbázis komplex struktúráját és azok kapcsolatrendszerét. A CPL, vagyis a Clean Paws Library, egy modern weboldal (később akár alkalmazás is), amely széleskörű funkcionalitást nyújt felhasználóinak. Célja, hogy kényelmes és interaktív módon tegye lehetővé a felhasználók számára könyvek rendelését otthonukból, könyvtárak legfrissebb híreinek és hirdetéseinek böngészését, valamint online pénzügyi tranzakciókat (pl. olvasójegy vásárlás, könyv kölcsönzési idő meghosszabbítása) végezzenek.

A Clean Paws Library rendszerét egy átfogó adatbázis hajtja végre, amely az egész platform működésének alapját képezi. Az adatbázisban tárolt információk között szerepelnek a könyvek adatai, felhasználói profilok, pénzügyi tranzakciók, könyvtárak tartalma és egyéb releváns információk. Az adatok struktúrált módon vannak tárolva, lehetővé téve a hatékony lekérdezéseket és a gyors adatelérést.

A Clean Paws Library adatbázisának kapcsolatrendszerét kifinomult logikai összefüggések alkotják. A könyvek, felhasználók és pénzügyi tranzakciók közötti kapcsolatok lehetővé teszik a rendszer számára, hogy dinamikusan reagáljon a felhasználói interakciókra. Például egy felhasználó könyvet rendelhet, majd az adatbázisban automatikusan frissül az ehhez kapcsolódó tranzakciói és az érintett könyv adatai.

Az adatbázisban való műveletek végrehajtására kifejlesztett funkciók, illetve azok hozzáférhetősége szorosan szabályozott. Ez biztosítja a biztonságot és az adatintegritást. A felhasználói élményt optimalizálva a Clean Paws Library adatbázisa rugalmas és skálázható, képes kezelni a növekvő adatforgalmat és az egyre bővülő felhasználói bázist.

A Clean Paws Library adatbázisa tehát nem csupán tárolóhelyként funkcionál, hanem egy dinamikus, interaktív és magas szintű funkcionalitást támogató rendszert hajt végre, mely lehetővé teszi a felhasználók számára, hogy kényelmesen és hatékonyan éljenek a könyvtári szolgáltatások széles skálájával.

## 2. Felhasználói felület

A felhasználói felület terve. Tagolás alpontokra szükség szerint.

A felhasználói felület egy HTML, Java programozási nyelveket használó internetes felület (weboldal) lesz. Ezért a felhasználónak nem kell mást tennie, csak rácsatlakozni a domain-re. A felület könnyen kezelhető (szép nagy formákkal, inkább szövegekkel történő kommunikáció, mint piktogramok), hogy a fiatalabb, de legfőképpen az idősebb osztály miatt.

A felhasználói felület részei:

**Bejelentkezési / regisztrációs felület:** Minden érkező látogatót ez a felület fog üdvözölni, itt lehet majd új fiókot létrehozni, valamint bejelentkezni, ha már van nekik. Azért van rá szükség, hogy a bejelentkezőket nyilván tudjuk tartani, és ne legyen a weboldal/adatbázis túlterhelve.



# Clean Paws Library

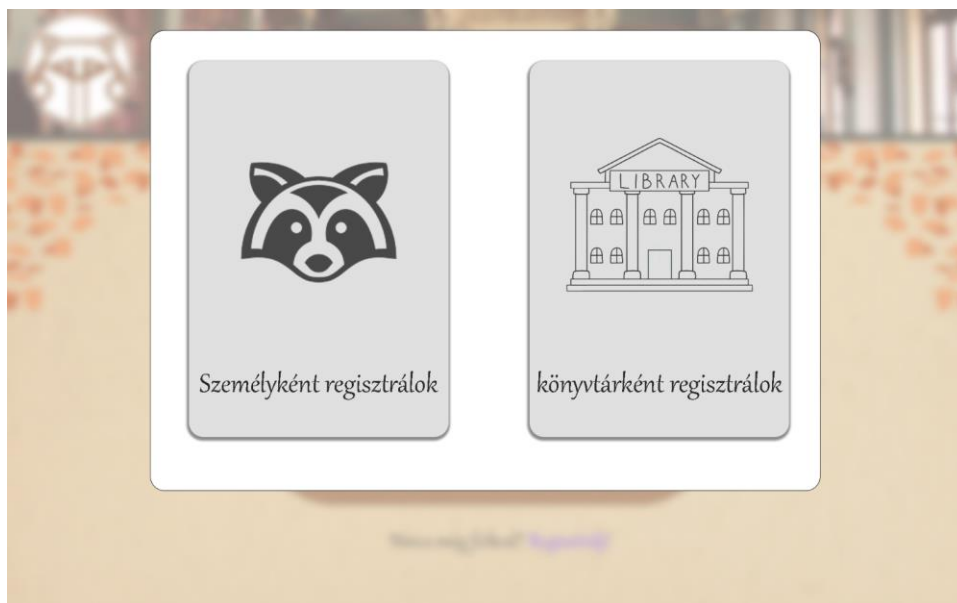
## Bejelentkezés

Nincs még fiókod? [Regisztrálj!](#)



# Clean Paws Library

## Regisztráció



**Kezdőlap:** A programunk közepe, innen érhető majd el minden funkció (persze csak bejelentkezés/regisztráció után).

A kezdőlap különbözni fog attól függően, hogy könyvtárként, vagy személyként regisztráltunk.

A funkciók elérése miatt van rá szükség, hogy a felhasználó könnyen tudjon tájékozódni.



**Kölcsönzés** Itt látja a felhasználó a függőben lévő kölcsönzéseit, valamint itt a plusz gombra kattintva lehet plusz időt adni a kölcsönzéshez. A felhasználó életének a megkönnyítéséért, illetve a kölcsönzésének a vizuális megtekintése miatt van ez a felület.



Mint a felhasználóknak, úgy a könyvtárnak is van egy kezdőlapja, amely a könyvtárosoknak segít elérni a legfontosabb funkciókat.



Könyvtári felhasználói felület: a könyvtár dolgozói itt láthatják és kezelhetik a könyvtárukhoz tartozó felhasználókat és a hozzájuk tartozó “folyamatokat” “kölcsönzés” olvasójegy”. Az egyszerűbb, könnyebb olvashatóság miatt van ez a felület.





# Clean Paws Library

Könyvek
Felhasználók
Statistikák

**Felhasználók**

Név:	Lakcím	Olvasójeggyel rendelkezik		Kölcsönzés	Lejárás
Kocsán László	Encs Petőfi utca 29.	Igen	Extra	Nincs	Nincs
Kovács Dániel	Miskolc Corvin utca 13.	Nem	Nincs	Harry Potter és az azkabani fogoly	2023.11.08

### 3. Adatmodellek

Ez a fejezet természetesen módosul, ha nem szükséges adatbázis az alkalmazáshoz. Abban az esetben azt kell leírni, hogy az adatok perzisztenciáját milyen más módon oldja meg a rendszer, illetve teljesen elmarad, ha a rendszer egyáltalán nem kezel perzisztens adatokat.

Ahhoz, hogy megértsük a program működését, mélyebben kell bemennünk az adatbázis-kezelés világába. A programunk valójában egy komplex adatbázis-kezelő egység, amely számos adatmodellezési és adatbázis-technológiai elvet alkalmaz a hatékony adatfeldolgozás és tárolás érdekében. Ebben a fejezetben részletesen tárgyaljuk az alkalmazott modelleket és az adatbázis-kezelőket, amelyeket célunk eléréséhez alkalmazunk.

Az adatmodellek, amelyekkel dolgozunk, a program szerves részét képezik, és ezek strukturált és szervezett formában tartalmazzák az alkalmazásunk által kezelt információkat. A modellek lehetnek hierarchikusak, kapcsolatokban állhatnak egymással, és komplex összefüggéseket tükrözhetnek az adatok között. Ezáltal lehetőséget biztosítanak a programnak a számunkra fontos adatok hatékony tárolására és lekérdezésére.

Az adatbázis-kezelők kiválasztása stratégiai döntés a program tervezése során. Különböző típusú adatbázisrendszerek léteznek, például relációs adatbázisok, dokumentum-orientált adatbázisok, grafikus adatbázisok stb. A kiválasztott adatbázis-kezelő technológiája befolyásolhatja az alkalmazás teljesítményét, skálázhatóságát és az adatok elérésének módját.

#### 3.1. Adatbázis kezelő kiválasztása

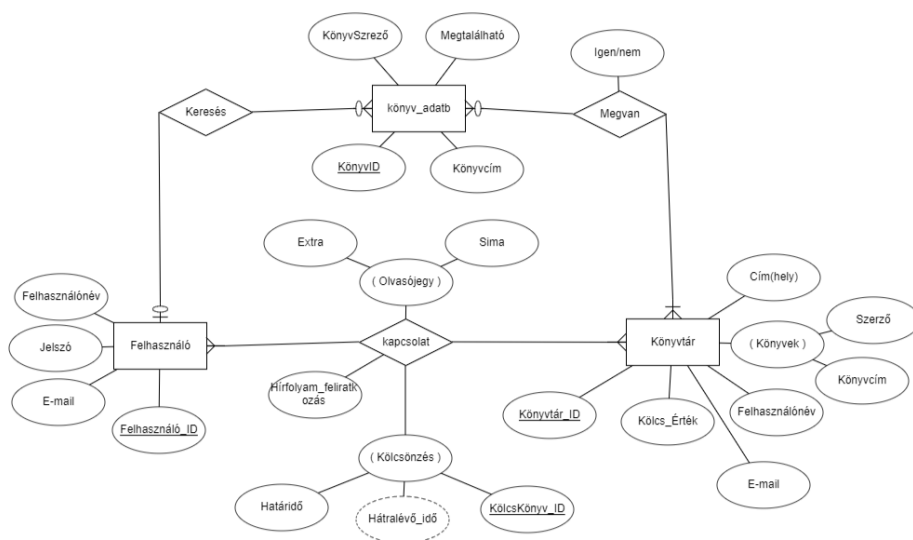
Itt kell véglegesen eldönteni, hogy az implementáció során milyen adatbázis kezelő használata szükséges, és milyen szabványos, esetleg egyedi technikával teremtjük meg a kapcsolatot az alkalmazás funkcionális részei és az adatbázis kezelő között.

Természetesen lehet az is a döntés, hogy tetszőleges adatbázis kezelő használatára készítjük fel a rendszert.

Adatbázis kezelőkből többféle áll rendelkezésre a piacon, viszont a csapatunk inkább az SQLite felé hajlik, ugyanis egyszerű kezelhetősége könnyebbé, és gyorsabbá teszi az adatbázis műveleteket, illetve eléggé könnyedén összemosható egy osztály funkcióival a parancsai, így osztályokból is tudjuk üzemeltetni. A hátránya az, hogy nem annyira “okos”, az adatbázis tulajdonságokat elég könnyű átcsapni, és még hibaüzenetet sem kapunk. Valamint felléphetnek sajátos hibák is. Jobb alternatíva lenne az Oracle, ami sokkal jobban tudja kezelni a hibákat, viszont annak a kezeléséhez a csapatnak nincs nagyon tapasztalata.

## 3.2. Szemantikai adatmodell

A szemantikai adatmodell megtervezése és ER diagrammal való szemléltetése ennek a pontnak a feladata.



Itt láthatjuk a teljes adatbázis relációs modelljét. Most nézzük meg, hogy egyes entitások miket is csinálnak.

**Felhasználó:** Neki van egy ID-je, mint minden jó adatbázisban, van egy felhasználóneve, illetve egy jelszava, amit, igen bizalmasan kell kezelni. Végül de nem utoljára egy E-mail címe, hogy minél könnyebben utol tudjuk érni.

**Könyvtár:** Neki már kicsit több tulajdonsága van, mint a felhasználónak. Ő már rendelkezik laccímmel, illetve egy kölcsönzési értéket megállapító tulajdonsággal. Valamint a könyvtárunkban található könyvek adatbázisával.

**Könyvtár-Felhasználói kapcsolat:** Ez a kapcsolat talán a legfontosabb az adatbázisban, ugyanis ez kezeli a két legfőbb osztály között a viszonyt, mégpedig (ahogy azt a relációs modellen is látjuk) a kölcsönzéseke, olvasójegyeket, illetve, hogy ki van feliratkozva a hírfolyamra.

**Könyv\_adatb:** Végül de nem utolsó sorban a teljes adatbázisa a könyveknek. Minden könyvnek itt is van egy ID-je egy szerzője, valamint egy címe.



### 3.3 Relációs adatmodell

A relációs adatmodellre konvertálás folyamata:

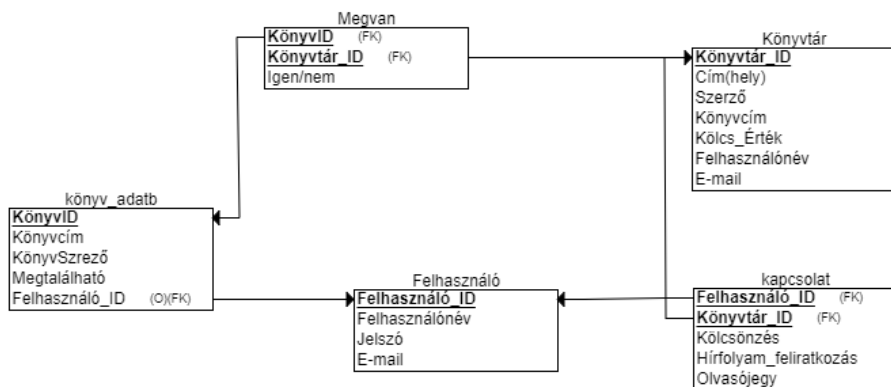
Leképezzük az adatmodellt egy kétdimenziós táblázatos adathalmazt úgy, hogy betartjuk a relációs adatmodell szabályait,

- Minden egyes táblázathoz tartozik egy azonosító
- A sorok és oszlopok metszetében található adatok egyértékűek ( atomiak), adatcsoportokat, tömböket nem tartalmazhatnak. Ezeket az elemi adatmezők
- Az oszlopokban található adatok azonos jellegűek, és egy előre definiált halmazból származnak
- Minden egyes oszlopnak egyedi neve van
- A táblázat minden sorában ugyanannyi adat található
- A táblázatban nem lehet két azonos sor
- A sorok és az oszlopok sorrendje nem lényeges

és a relációs adatmodell leírása: relációs modellünkben jól látható, hogy a programban szereplő főbb osztályok, hogy és milyen kapcsolatban állnak egymással az adatbázison belül / melyek azok az adatok, amelyek összekapcsolják őket,

Például: “Megvan” kapcsolat mutatja nekünk, hogy az adott könyvtárba az adott ID vel rendelkező könyv jelenleg elérhető vagy sem

vagy “kapcsolat” kapcsolat azt, hogy az adott felhasználó milyen szolgáltatásokat vesz igénybe az adott könyvtártól.



### 3.4. Az adatbázis kezelővel kapcsolatot tartó osztályok

Az adatbázis és a program funkcionális része közötti kapcsolat fenntartását a CPL osztály látja el. Ennek az osztálynak a kulcsfontosságú szerepe abban áll, hogy összeköti az alkalmazás magját, az adatok kezelését és a kimeneti eredmények előállítását. A CPL osztály gyakorlatilag a rendszer agya, amely felügyeli és irányítja az információáramlást a különböző részek között.

A CPL osztály nem csupán adatokat ír ki fájlba vagy listába, hanem felelős az adatbázisok manipulálásáért is. Az adatbázisok módosítása során a CPL osztály különböző metódusai biztosítják az adatok konzisztenciáját és integritását. Emellett a CPL osztály felelős az új adatok felvételéért és azok rendszerbe történő integrálásáért is.

Mint a rendszer root-osztálya, a CPL osztály magába foglalja az alkalmazás legmagasabb szintű vezérlését és koordinációját. A funkcionális részek közötti együttműködési pontként szolgál, biztosítva, hogy az adatok helyesen és hatékonyan áramoljanak az alkalmazáson belül. A tudományos megközelítésének köszönhetően a CPL osztály tervezése és működtetése optimalizált, figyelembe véve az algoritmusok hatékonyságát és az adatkezelés legújabb fejlesztéseit.

Ezenkívül a CPL osztály lehetőséget kínál az alkalmazásnak az adaptációra és bővítésre, mivel rugalmas interfészt biztosít az új funkciók hozzáadásához és az alkalmazás fejlesztéséhez. A tudományos alapokon nyugvó tervezése révén a CPL osztály képes alkalmazkodni az új technológiai trendekhez és változásokhoz, így biztosítva az alkalmazás hosszú távú élettartamát és hatékonyságát.